Fuzzy Pooling

Aditya Suryawanshi
210150004

Table of contents

01

Introduction

Introduction

- CNN Impact: Convolutional Neural Networks (CNNs) have revolutionized computer vision, excelling in tasks like image classification, object detection, and segmentation.
- Role of Pooling Layers: Pooling layers in CNNs reduce feature map dimensions, lowering computational demands.
- Limitations of Traditional Pooling: Common pooling methods (max and average pooling) often ignore uncertainty in visual data.(Noise,Color ambiguities, blurring and resolution issues,occlusion,etc)
- Fuzzy Logic in Neural Networks: Applying fuzzy logic to CNNs enables more effective handling of imprecise or vague data, beneficial in domains like image processing.

02

Objective

To develop and evaluate a novel fuzzy pooling operation for CNN architectures, aimed at managing the uncertainty of feature values and improving classification performance.
This new pooling method is designed to serve as a drop-in replacement for traditional pooling layers, with the goal of preserving important feature details across various image datasets.

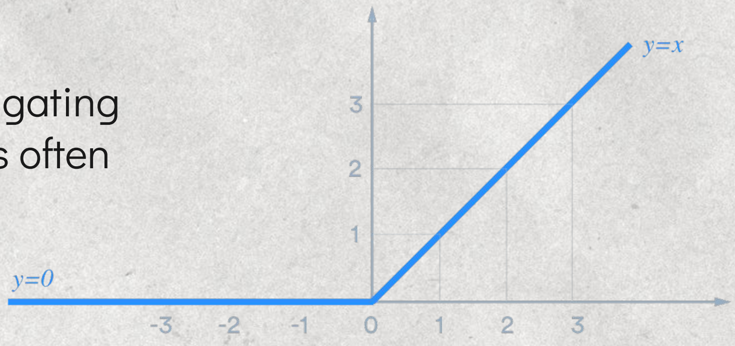
03

Methodology

Methodology

Choosing Activation Function

<u>ReLU</u>: $\text{ReLU}(x) = \max(0, x)$
Preferred over sigmoid due to simplicity and mitigating vanishing gradients which deep neural networks often suffer from.



<u>Capped ReLU</u>: $\text{ReLU}(x, r\max) = \min(\max(x, 0), r\max)$
Introduces $r\max$ to cap output, keeping it within a range while retaining ReLU's benefits.

Methodology

Patch For Pooling

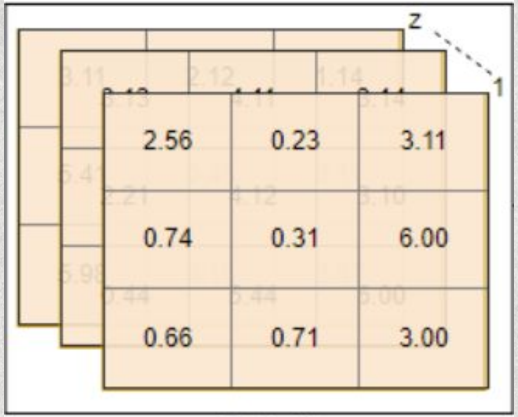
Volume β: Size w×h×z, containing z feature maps βn of size w×h.

Pooling Process: Uses a k×k window (e.g., k=3) with stride σ=2, halving width and height.

Volume Patches: Extracted with stride σ, forming spatial patches p_n from each feature map β_n.

Patch Count: $c = (w - k + 2t_w)*(h - k + 2t_h)/(2\sigma + 2)$

where $t_w = (\sigma - 1)(w - \sigma + k)/2$ and $t_h = (\sigma - 1)(h - \sigma + k)/2$ are the zero padding values used in the patch extraction process along the width and height axes, respectively.



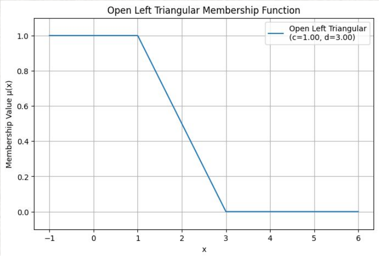
Patch (p)
(k×k×z)

Methodology

Membership Functions

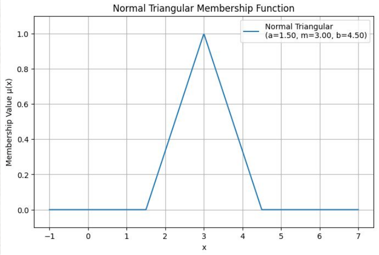
The first membership function is defined as:
Where d = rmax/2 and c = d/3 .

$$\mu_1(p_{i,j}^n) = \begin{cases} 0 & \text{if } p_{i,j}^n > d \\ \frac{d - p_{i,j}^n}{d - c} & \text{if } c \le p_{i,j}^n \le d \\ 1 & \text{if } p_{i,j}^n < c \end{cases}$$



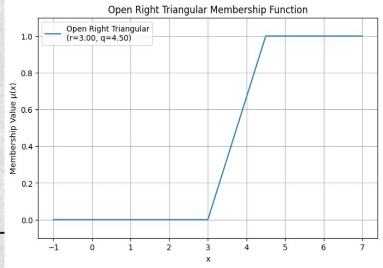
The second membership function is defined as:
Where a = rmax/4 , m =rmax/2 and b = m + a.

$$\mu_2(p_{i,j}^n) = \begin{cases} 0 & \text{if } p_{i,j}^n \le a \\ \frac{p_{i,j}^n - a}{m - a} & \text{if } a \le p_{i,j}^n \le m \\ \frac{b - p_{i,j}^n}{b - m} & \text{if } m < p_{i,j}^n < b \\ 0 & \text{if } p_{i,j}^n \ge b \end{cases}$$



The third membership function is defined as:
Where a = rmax/4 , m =rmax/2 and b = m + a.

$$\mu_3(p_{i,j}^n) = \begin{cases} 0 & \text{if } p_{i,j}^n < r \\ \frac{p_{i,j}^n - r}{q - r} & \text{if } r \le p_{i,j}^n \le q \\ 1 & \text{if } p_{i,j}^n > q \end{cases}$$

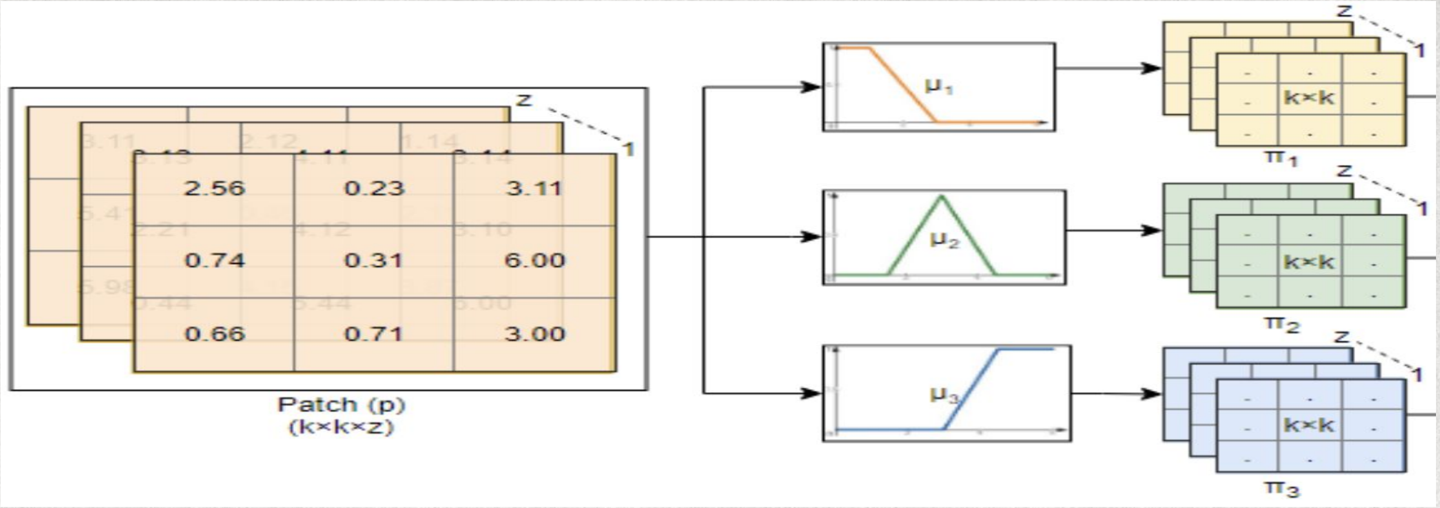


Methodology

Fuzzification

For each patch p_n, where $n = 1, \dots, z$, a fuzzy patch π_v^n is defined as:

$$\pi_v^n = \mu_v(p^n) = \begin{pmatrix} \mu_v(p_{1,1}^n) & \cdots & \mu_v(p_{1,k}^n) \\ \vdots & \ddots & \vdots \\ \mu_v(p_{k,1}^n) & \cdots & \mu_v(p_{k,k}^n) \end{pmatrix}$$



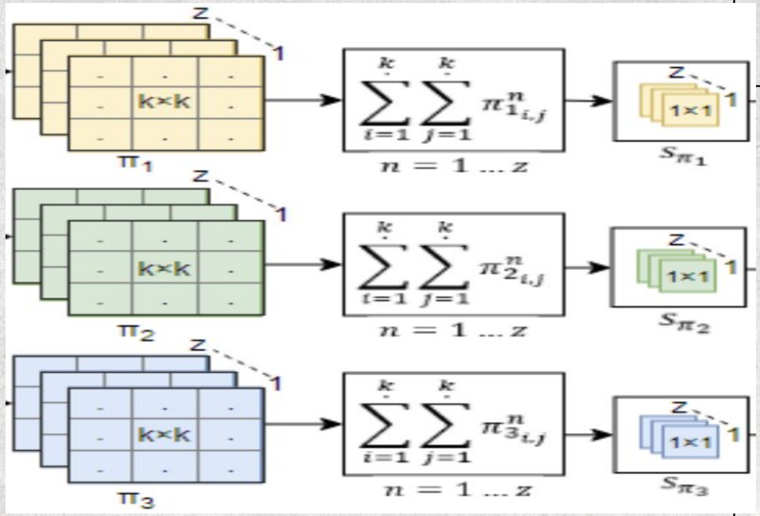
Methodology

Score formulation for each fuzzy patch.

Pooling begins with the spatial aggregation of the values of the fuzzy patch, using the fuzzy algebraic sum operator Σ, as Follows The value of each $s\pi v\ n$ is considered as a score quantifying the overall membership of p_n to $\tilde{y}_v$.

$$s^n_{\pi_v} = \sum_{i=1}^{k}\sum_{j=1}^{k} \pi^n_{v,i,j}, \quad n = 1, \ldots, z.$$

The values are considered as a score quantifying the overall membership of p_n to $\tilde{y}_v$.

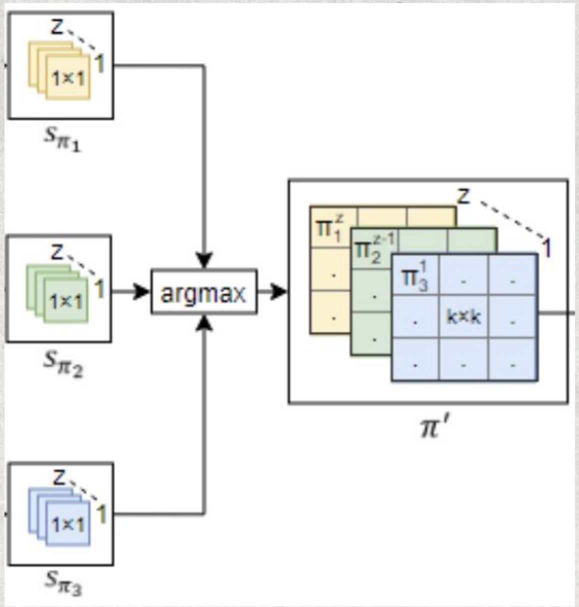


Methodology

Final Fuzzy Volume Patch

Based on these scores, for each volume patch p, a new fuzzy volume patch π′ is created by selecting the spatial fuzzy
patches π_n for v = 1, . . . ,V that have the largest scores s_n, i.e.,

$$\pi' = \{\pi'_n = \pi_n \mid v = \arg\max(s_n),\, n = 1, 2, \ldots, z\}$$

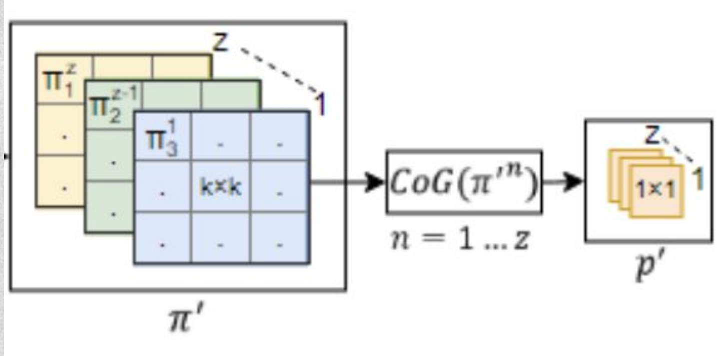


Methodology

Defuzzification

The dimensionality of each patch is then reduced by defuzzification using the Center of Gravity (CoG):

$$p'_n = \frac{\sum_{i=0}^{k}\sum_{j=0}^{k}\left(\pi'_{n,i,j}\cdot p_{n,i,j}\right)}{\sum_{i=0}^{k}\sum_{j=0}^{k}\pi'_{n,i,j}}$$

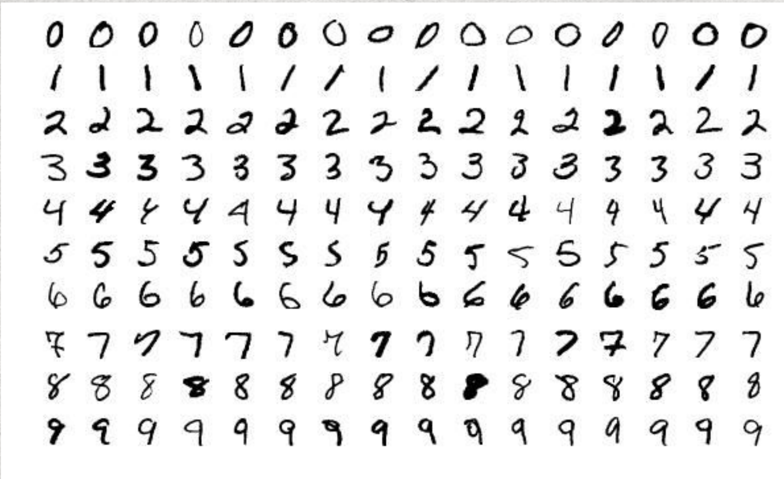


04

Results And Analysis

Dataset

Classification performance was assessed using MNIST which contains 70,000 grayscale images of handwritten digits (0-9), sized 28 × 28 pixels. comprising 60,000 training and 10,000 test images. All the experiments were conducted using the training and testing subsets provided by the datasets, which are class-balanced.

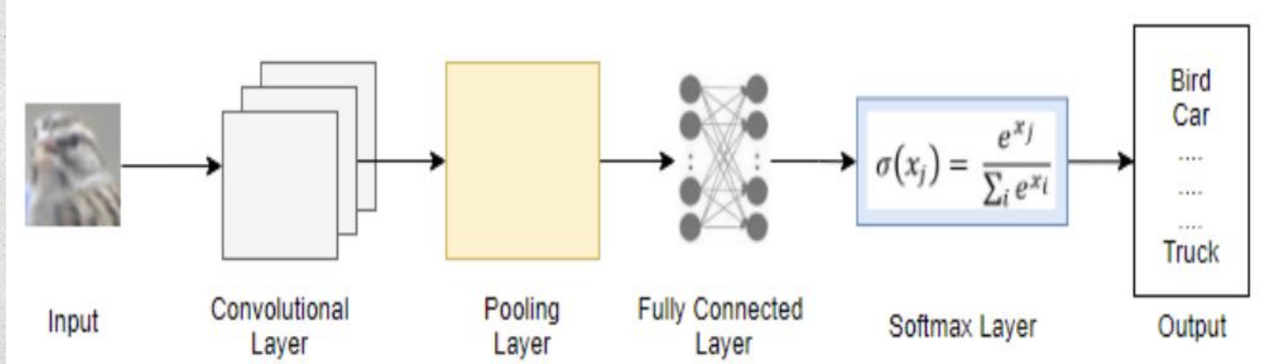


Sample images from MNIST dataset.

The paper also implements it on two other datasets namely, Fashion-MNIST and CIFAR-10 .

Architecture

- To minimize hyper-parameter bias, we evaluated the proposed pooling method using the LeNet baseline CNN.
- LeNet has fewer parameters than modern architectures, helping isolate the pooling methods performance impact.
- Training used Stochastic Gradient Descent with a batch size of 32 images.
- No data preprocessing or augmentation was applied to focus solely on the pooling layers impact on classification.



Standard LENET Architecture

Results

From the paper :

TABLE I
COMPARATIVE ACCURACY RESULTS OF THE PROPOSED TYPE-1 FUZZY POOLING METHODOLOGY ON MINST DATASET [32]

Methodology	Classification Accuracy
Max Pooling	88.48%
Average Pooling	94.06%
RegP [16]	95.46%
Type-2 Fuzzy Pooling [17]	94.40%
Proposed	**98.56%**

TABLE II
COMPARATIVE ACCURACY RESULTS OF THE PROPOSED TYPE-1 FUZZY POOLING METHODOLOGY ON CIFAR-10 DATASET [34]

Methodology	Classification Accuracy
Max Pooling	70.73%
Average Pooling	74.83%
RegP [16]	75.44%
Type-2 Fuzzy Pooling [17]	27.92%
Proposed	**78.35%**

TABLE III
COMPARATIVE ACCURACY RESULTS OF THE PROPOSED TYPE-1 FUZZY POOLING METHODOLOGY ON FASHION-MNIST DATASET [33]

Methodology	Classification Accuracy
Max Pooling	84.28%
Average Pooling	85.90%
RegP [16]	86.41%
Type-2 Fuzzy Pooling [17]	N/A
Proposed	**88.57%**

My Implementation :

Methodology	Classification Accuracy
Max Pooling	98.69
Average Pooling	98.10
Fuzzy Pooling	98.80

The model was trained and tested on MNIST dataset. It shows and improvement in accuracy over average pooling and almost better performance than max pooling . With better system configurations and larger data size to train on the results for fuzzy pooling further improve .(Also in data that has high uncertainties)

05

Conclusion

Conclusion

- Novel Contribution: Introduced a new fuzzy pooling method for CNNs to handle feature value uncertainty.
- Performance Improvement: Demonstrated that fuzzy pooling significantly enhances CNN classification performance over existing pooling methods.
- Feature Preservation: Proven on standard image datasets that fuzzy pooling better retains crucial features in pooled areas.

Thanks!